MAMUT: A Novel Framework for Modifying Mathematical Formulas for the Generation of Specialized Datasets for Language Model Training

Anonymous authors
Paper under double-blind review

Abstract

Mathematical formulas are a fundamental and widely used component in various scientific fields, serving as a universal language for expressing complex concepts and relationships. While state-of-the-art transformer models excel in processing and understanding natural language, they encounter challenges with mathematical notation, which involves a complex structure and diverse representations. This study focuses on the development of specialized training datasets to enhance the encoding of mathematical content. We introduce Math Mutator (MAMUT), a framework capable of generating equivalent and falsified versions of a given mathematical formula in LaTeX notation, effectively capturing the mathematical variety in notation of the same concept. Based on MAMUT, we have generated four large mathematical datasets containing diverse notation, which can be used to train language models with enhanced mathematical embeddings.

1 Introduction

Mathematical formulas are a fundamental and widely used component in various scientific fields, serving as a universal language for expressing complex concepts and relationships. Their context-dependent symbols, nested operations, and diverse notations pose distinct challenges for machine learning models due to their symbolic and structural differences from natural language (Zanibbi et al., 2020; Peng et al., 2021).

Despite the success of transformer-based language models (Vaswani et al., 2017) in natural language tasks, they encounter challenges in comprehending mathematical notation (Hendrycks et al., 2021; Gong et al., 2022; Petersen et al., 2023; Dao & Le, 2023; Shen et al., 2023; Reusch et al., 2024; Qiao et al., 2024). These challenges stem from the complex formula structure, diverse formula representations, and ambiguous implicit semantics (Peng et al., 2021). For example, $x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$ involves nested operations, while different notations, such as $\frac{x}{y}$, x/y, $x \div y$, and $x \cdot y^{-1}$ can represent the same mathematical relationship, alongside the contextual meanings of symbols (e.g., i as an index or imaginary unit) further complicate the understanding. These difficulties highlight the need for rich, specialized datasets to train models for

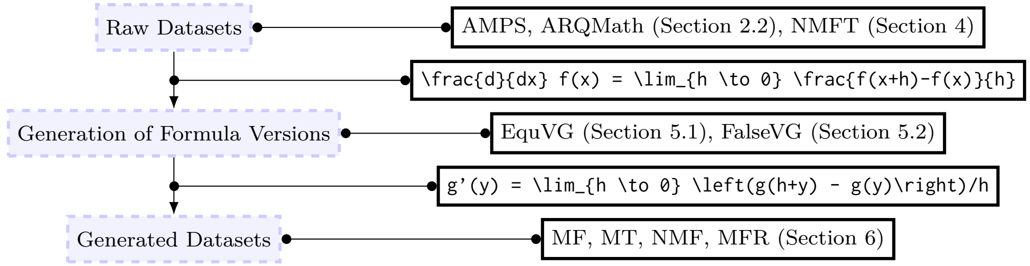

Raw Datasets
AMPS, ARQMath (Section 2.2), NMFT (Section 4)
\frac{d}{dx} f(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}
Generation of Formula Versions
EquVG (Section 5.1), FalseVG (Section 5.2)
g'(y) = \lim_{h \to 0} \left(g(h+y) - g(y)\right)/h
Generated Datasets
MF, MT, NMF, MFR (Section 6)

Figure 1: MAMUT: Modifying formulas to generate large and diverse mathematical datasets.

Dataset	**Description**	**Example(s)**
Mathematical Formulas (MF)	Mathematical formulas with high variance	$x \cdot x^N = x^{1+N}$ $(a-b)/(b*a) = -1/a + \frac{1}{b}$
Mathematical Texts (MT)	Texts combining natural language and mathematical formulas	Identify $\sum_{n=0}^{\infty}(y_n - L)$ where $y_{n+1} = (1+y_n)^{\frac{1}{3}}$ and $L^3 = L+1$. Let $y > 2$ and let $f(y) = (1+y)^{\frac{1}{3}}$. Let $f^n(y)$ be the n th iterate of $f(y)$. Let L be ...
Named Mathematical Formulas (NMF)	High variance formulas of famous named identities	Name: Pythagorean Thm., Formula: $c^2 = b^2 + a^2$ Name: Binomial Formula, Formula: $(\alpha + z)^2 = z^2 + \alpha^2 + 2 \cdot \alpha \cdot z$
Mathematical Formula Retrieval (MFR)	Pairs of formulas with labels indicating identical or different mathematical concepts	Formula 1: $1 \cdot 2 \cdot 3 \cdot \ldots \cdot n = n!$, Formula 2: $m! := \prod_{k=1}^{m} k$, Label: Equivalent Formula 1: $a^2 + b^2 = c^2$, Formula 2: $a^2 + 2^b = c^2$ Label: Not Equivalent

Table 1: Overview of generated datasets. The examples are shown as rendered LaTeX.

mathematical content. However, existing datasets face scalability constraints due to expert curation or lack diversity in problem types and notation.

To address the need, we propose a framework, Math Mutator (MAMUT), for generating high-quality and diverse mathematical formulas to enhance the training and comprehension capabilities of mathematical language models. MAMUT allows for the creation of mathematically equivalent formulas (EquVG) and challenging non-equivalent ones that appear similar (FalseVG). This includes random alterations in variable and function identifiers and variations in LaTeX notation that leverage mathematical properties such as commutativity and symmetry. Additionally, we extend this approach to text containing mathematical LaTeX notation, ensuring consistent changes in identifiers and notation styles across textual contexts. We apply MAMUT to generate four datasets (see Figure 1 and Table 1) designed for the training of mathematical language models, e.g., for further mathematical pre-training on equation completion tasks.

2 Related Work

This section covers language models, datasets, and data augmentation techniques in mathematical contexts.

2.1 Mathematical Language Models

The success of transformer-based models (Vaswani et al., 2017), such as BERT (Devlin et al., 2019), led to the development of domain-specific models, including SciBERT for scientific texts (Beltagy et al., 2019) and CodeBERT for programming (Feng et al., 2020). These models improve over general-purpose models by training on domain-specific data. Similarly, specialized models have been developed for mathematics, such as MathBERT (Peng et al., 2021), MathBERTa (Novotný & Štefánik, 2022), and others (Reusch et al., 2022; Liu et al., 2023; Shao et al., 2024). They typically employ additional mathematical tokens and were pre-trained on mathematical datasets (see Section 2.2).

A key application of mathematical language models is in Mathematical Information Retrieval (MIR) (Dadure et al., 2024; Zanibbi et al., 2025), where the goal is to retrieve relevant documents based on a user's query, where both may contain mathematical content (see Table 2 for examples). Traditional MIR systems rely on keyword matching or simple embeddings (Kim et al., 2012; Greiner-Petter et al., 2020), while more sophisticated techniques leverage explicit mathematical knowledge, such as operator trees or formula unification (Kristianto et al., 2016; Mansouri et al., 2019; 2022b; Aizawa & Kohlhase, 2021; Peng et al., 2021). Transformer-based models offer new possibilities for MIR by addressing key challenges such as integrating natural and mathematical language, directly processing LaTeX input, and handling diverse notations. As a

Query	**Relevant Documents**	**Not Relevant Documents**
$(a+b)^2 = a^2 + 2ab + b^2$	$a^2 + 2ab + b^2 = (a+b)^2$	$(a+b)^2 + a^2 = 2ab + b^2$
	$(c+d)^2 = c^2 + 2cd + d^2$	$(a+b)^2 = a^2 + 2ab + a^2$
	$(a+b)^2 = a^2 + b^2 + 2ab$	$(a+b)^2 = a^2 + b^2$
$a^2 + b^2 = c^2$	$c^2 = a^2 + b^2$	$a^2 + b^2 + c^2$
Pythagorean Theorem	$a^2 + b^2 = c^2$	Pythagorean Identity
$\sum_{n=1}^{\infty} \frac{1}{n}$	$\sum_{k=1}^{\infty} k^{-1}$	$\sum_{n=1}^{\infty} \frac{1}{n^2}$
$f'(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}$	$\frac{d}{dz} g(z) = \lim_{d \to 0} \frac{g(z+d)-g(z)}{d}$	$\lim_{x \to 0} f(x) = 0$

Table 2: Examples for MIR queries including relevant and not relevant documents. Note that *Pythagorean Identity* ($\sin^2(x) + \cos^2(x) = 1$) is not relevant for the query *Pythagorean Theorem* ($a^2 + b^2 = c^2$).

result, mathematical language models have been successfully adapted to MIR (Novotný & Štefánik, 2022; Reusch et al., 2022; Zhong et al., 2023).

Despite their promising performance, specialized mathematical models still face challenges in understanding mathematical notation (Gong et al., 2022; Shen et al., 2023), especially when it comes to handling variable names and recognizing mathematical equivalence beyond superficial textual similarities (Reusch et al., 2024). This motivates the creation of specialized datasets that reflect the unique roles of variables and aim to improve mathematical modeling.

2.2 Mathematical Datasets

The need for mathematical models has led to the development of various collections aimed at enhancing and evaluating language model capabilities in context of mathematics. Manually curated datasets like MATH (Hendrycks et al., 2021), GSM8K (Cobbe et al., 2021), and MathOdyssey (Fang et al., 2024) test problem-solving skills but are typically small, and reliant on expert input. Synthetically generated datasets like the Mathematics Dataset (Saxton et al., 2019), AMPS (Hendrycks et al., 2021), and HARDMath (Fan et al., 2024) offer scalability but may lack diversity in problem topics, as they rely on generation rules, although they can produce a wide variety of similar but distinct problems (with changing numbers, symbols, ...), which can be beneficial for models learning to generalize across formula representations. Datasets like NTCIR (Zanibbi et al., 2016) and ARQMath (Mansouri et al., 2022a), sourced from the existing repositories arXiv, Wikipedia, and the Mathematical Stack Exchange, provide a broad range of real-world mathematical problems. However, they lack controlled variations of specific formulas, an important aspect for training MIR models to classify whether two symbolic representations are mathematically equivalent.

2.3 Mathematical Data Augmentation Techniques

Recent advancements in mathematical data augmentation have introduced various innovative methods aimed at enhancing the diversity and depth of training materials. InfinityMath (Zhang et al., 2024) utilize GPT-4 (Achiam et al., 2023) to transform specific mathematical problems into generic templates. These templates can then generate multiple variations of the original problem, altering numerical values or structural complexity, thereby increasing the dataset's variety. Similarly, Li et al. (2024) propose a method to formalize mathematical problems written in natural language, alter the difficulty by adjusting the problem's operations, and then informalize these changes back into natural language using GPT-4, preserving mathematical integrity across different levels of complexity. MathGenie (Lu et al., 2024) augments step-by-step solutions by generating modified candidate solutions with a Llama model (Touvron et al., 2023) with verified correctness, and then back-propagating these solutions to a modified question. You et al. (2024) augment data by applying different strategies, including rephrasing and reorganization with LLM, and question alteration.

	Natural Language	**Mathematical Language**
Purpose	General human communication, including opinions and feelings	Precise description of mathematical concepts
Vocabulary	Large set of words (language dependent), sometimes with ambiguous meaning (e.g., *love*, *happy*, *data*)	Small set of well-defined symbols (e.g., x, $+$, $\mathbb{R}$, sin, $\forall$, ∞, $\int$) and terms (e.g., *Eigenvalue*, *Derivative*, *Field*) with precise meanings
Grammar	Rather flexible	Strict rules
Clarity	Often imprecise, open to interpretation	Single, unambiguous interpretation
Evolution	Evolves over time naturally, new words, phrases, and idioms emerge or disappear	Evolves slower, changes are introduced by mathematicians and are backward compatible
Writing Style	Linear structure in sentences and paragraphs using standard formats	Requires specialized formats (e.g., LaTeX) to represent complex notation in a structured way

Table 3: Comparison of natural and mathematical language.

These approaches primarily focus on diversifying problem content rather than varying mathematical notation (e.g., $(a+b)^2 = a^2 + 2ab + b^2$ vs. $(x+y)^2 = x^2 + y^2 + 2yx$). In contrast, Reusch et al. (2024) explore variable renaming in training data to prevent models from taking shortcuts in problem-solving, such as relying only on variable overlap. Building on this idea, our study enhances mathematical formula diversity through substitutions and other techniques.

3 Natural and Mathematical Language

Mathematical language differs fundamentally from natural languages such as English, German, or Chinese. While natural language is used for general communication and often conveys subjective information, mathematical language serves a highly specialized purpose to precisely describe mathematical topics, such as definitions, theorems, and proofs. It consists of both symbolic expressions (e.g., $a^2+b^2 = c^2$ and $\int_a^b \sin(x), dx$) and specialized terminology (e.g., *derivative* and *eigenvalue*). Despite their differences, natural and mathematical languages share some structural similarities. Both use symbols arranged in a syntax that conveys meaning, and both can be represented in textual form. However, there are some key differences (Ilany et al., 2010; Scarlatos & Lan, 2023) summarized in Table 3. A crucial challenge for mathematical language models is *symbol abstraction*. In mathematical expressions, certain symbols act essentially as wildcards and can be replaced without changing the mathematical meaning. These symbols are either *variables* (e.g., x, α, A) or *generic functions* (e.g., f, g or φ), i.e., functions not tied to a specific mathematical object (e.g., Euler's Gamma function $\Gamma(z)$). For example, a model should recognize that the first binomial formula,

$$(a+b)^2 = a^2 + 2ab + b^2, \tag{1}$$

is equivalent to $(c+d)^2 = c^2+2cd+d^2$, despite different variable names. In contrast, $(a+b)^2 = a^2+2ab+\boldsymbol{a^2}$ uses the same variables but in a mathematically non-derivable way. Likewise, the modified formula $(a+b)^2 = a^2 + b^2 + 2ab$ appears different from Eq. (1), but it is, in fact, mathematically equivalent.

Another important aspect is the structure of mathematical formulas. Consider the two formulas 2^x and x^2. Assuming a character-wise LaTeX tokenization, both formulas use the same tokens but in a different order. A model should not treat x^2 as equivalent to 2^x (but instead to $x \cdot x$). These structural variations highlight the need for a model that goes beyond simple token matching and actually understands mathematical meaning. Transformer language models (Vaswani et al., 2017) have shown their capabilities in modeling natural language, hence, it is worth exploring their potential to precisely capture mathematical language.

Names	Factorial, Definition of a factorial
Version 1	$n! = 1 \cdot 2 \cdot \ldots \cdot n$
Version 2	$n! = \prod_{i=1}^{n} i$
Version 3	$\forall n \in \mathbb{N} : (n+1)! = (n+1) \cdot n!,\ 0! = 1$
Version 4	For any natural number n, we have $n!$ is defined as $n! := \prod_{i=1}^{n} i$.
Version 5	For any natural number n, $n!$ can be obtained by multiplying all natural numbers from 1 to n together.
Similar Formula	Binomial Coefficient Formula
False Version 1	$n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot n$
False Version 2	$\forall n \in \mathbb{N} : (n+1)! = (n-1) \cdot n!,\ 0! = 0$
Falsifying Replacements	$\prod \to \sum$, $\mathbb{N} \to \mathbb{R}$, “natural“ → “real“

Table 4: Example entry for the definition of a factorial from the NMFT dataset (partially).

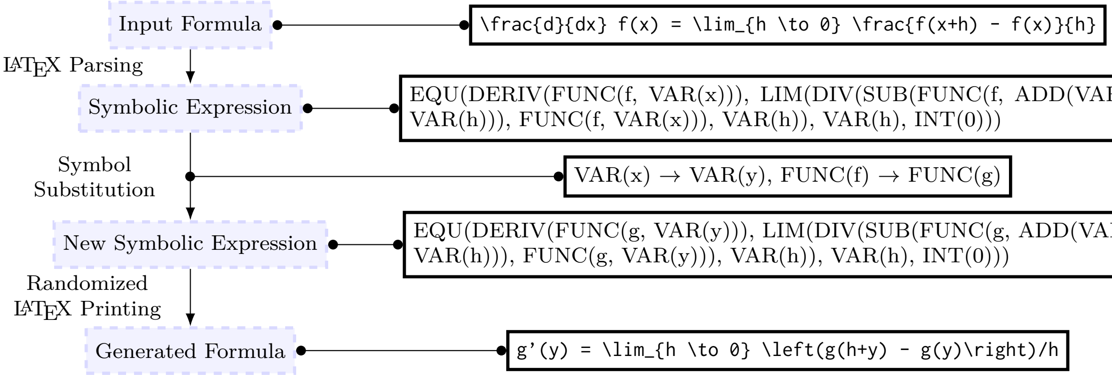
4 Named Mathematical Formula Templates (NMFT)

Previous datasets provide formulas and mathematical texts with significant variance across a wide range of mathematical topics. For the purpose of our proposed data augmentation methods introduced in the next section, it is necessary to parse formulas into symbolic expressions. Real-world datasets contain formulas with various notations, some of them might be parsed incorrectly, or the parsing even fails completely. Therefore, we created a dataset consisting of only a few but high-quality parsable formulas. This dataset includes 71 well-known mathematical identities that are easily recognizable and associated with one or multiple names. For example, $a^2 + b^2 = c^2$ represents the Pythagorean theorem, while $(a+b)^2 = a^2 + 2ab + b^2$ represents the first binomial formula (Eq. 1). Since mathematical formulas are associated with its name, we call this dataset Named Mathematical Formula Templates (NMFT), as the formulas serve as templates for deriving modified versions. An example entry can be found in Table 4, and Table 6 lists all identities.

Each identity provides multiple representations, such as $\forall a, b \in \mathbb{R} : (a+b)^2 = a^2 + 2ab + b^2$ as another, more detailed version of the first binomial formula. Additionally, some representations are provided as descriptive text, e.g., *“In a right-angled triangle with side lengths a, b, and c, where c represents the length of the hypotenuse, the equation $a^2 + b^2 = c^2$ holds true”*. Others paraphrase formulas textually, e.g., *“$a^2 + b^2$ is equal to c^2”*, reinforcing associations between the equals sign = and *“equals”*. These textual versions intentionally exclude the name of the identity to make MIR tasks harder, where the name serves as the query. For each provided identity version, the variables and function symbols are explicitly given to assist the parsing and version generation. To enhance the generation of challenging falsified versions, similar-looking formulas are provided (e.g., the first binomial formula for the Pythagorean theorem, as both identities contain multiple powers of two), or hints to falsify any given representation by a string replacement (e.g., removing *“right-angled”* to falsify the previous descriptive example of the Pythagorean theorem). The descriptive text versions have been partially generated by using GPT-3.5 (Brown et al., 2020) and manually verified for validity. Typically, we call entries of NMFT and of datasets generated from it *formulas*, but both, pure mathematical formulas and textual descriptions of formulas are meant.

5 Math Mutator (MAMUT)

Our goal is to create high-quality, large, and diverse mathematical datasets to enhance mathematical modeling. We introduce Math Mutator (MAMUT), a framework consisting of two core algorithms designed to generate both equivalent and falsified versions of a given formula. The first algorithm, Equivalent Version



```
\frac{d}{dx} f(x) = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}
EQU(DERIV(FUNC(f, VAR(x))), LIM(DIV(SUB(FUNC(f, ADD(VAR(x), VAR(h))), FUNC(f, VAR(x))), VAR(h)), VAR(h), INT(0)))
VAR(x) → VAR(y), FUNC(f) → FUNC(g)
EQU(DERIV(FUNC(g, VAR(y))), LIM(DIV(SUB(FUNC(g, ADD(VAR(y), VAR(h))), FUNC(g, VAR(y))), VAR(h)), VAR(h), INT(0)))
g'(y) = \lim_{h \to 0} \left(g(h+y) - g(y)\right)/h
```

Figure 2: Visualization of the EquVG algorithm.

Generation (EquVG), presented in Section 5.1, automatically generates various versions of a given formula, expanding the training data and enabling the model to learn math-specific language rules, such as treating variables as placeholders that can be substituted without changing the validity of an expression. The second algorithm, Falsified Version Generation (FalseVG), introduced in Section 5.2, slightly modifies formulas slightly to create mathematically not equivalent versions of the original formula, offering challenging negative examples for MIR tasks.

5.1 EquVG: Variations of Mathematical Formulas

The key idea of this section can be summarized as follows: Given a mathematical formula, our aim is to generate mathematically equivalent variations of this formula, called *equivalent versions*. For instance, consider the formula

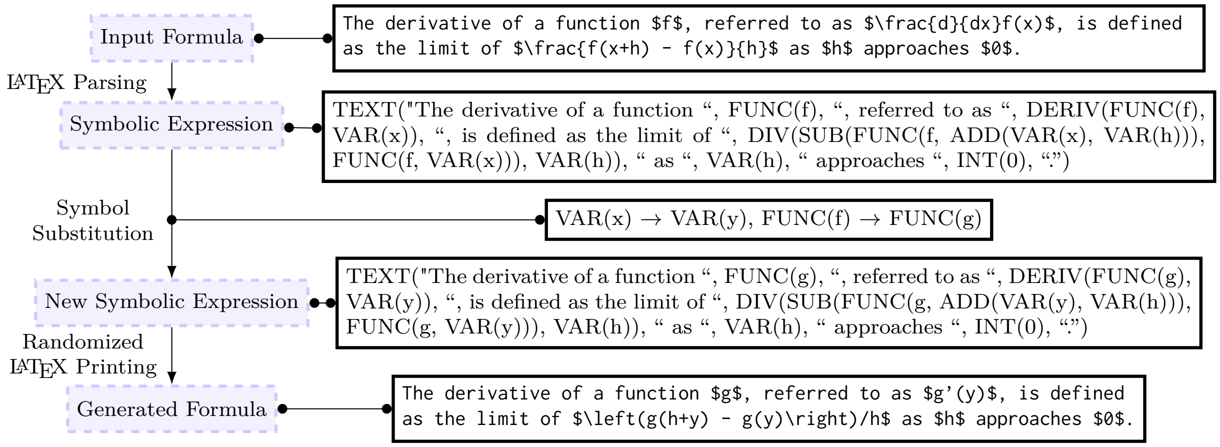
$$(a+b)^2 = a^2 + 2 \cdot a \cdot b + b^2. \tag{2}$$

In this context, we observe that all the following formulas describe the same mathematical relationship, namely the first binomial formula:

$$(b+a)^2 = a^2 + b^2 + 2 \cdot b \cdot a, \tag{3}$$
$$(a+b)^2 = a \cdot a + 2 \cdot a \cdot b + b^2, \tag{4}$$
$$a^2 + 2 \cdot a \cdot b + b^2 = (a+b)^2, \tag{5}$$
$$(c+d)^2 = c^2 + 2 \cdot c \cdot d + d^2, \tag{6}$$
$$(\lambda + Z)^2 = \lambda^2 + 2 \cdot \lambda \cdot Z + Z^2. \tag{7}$$

Equation (3) can be derived from Eq. (2) by applying both, additive and multiplicative commutativity. In Eq. (4), the exponentiation a^2 is replaced by its definition $a \cdot a$. Since equality is a symmetric relation, equations remain valid after interchanging the sides, as done in Eq. (5). The final two equations can be obtained from Eq. (2) by substituting variables. This section is dedicated to the automated generation of such equivalent versions. Note that for a complete mathematical expression, it would be necessary to specify the range of values (e.g., of variables) for which the statement holds. For example, a complete expression of Eq. (2) could be $\forall a, b \in \mathbb{R} : (a+b)^2 = a^2 + 2 \cdot a \cdot b + b^2$. However, in practical applications like MIR systems, the shortened version Eq. (2) may also be used, for the sake of simplicity. Therefore, the somewhat less precise mathematical formulations in Eqs. (2)-(7) are often sufficient.

The complete workflow of our proposed Equivalent Version Generation (EquVG) algorithm is depicted in Figure 2. The input consists of a formula written in LaTeX format. To implement transformations, as seen in Eqs. (2)-(7), we can identify two steps: the substitution of symbols and the modification of the mathematical notation. For both of these purposes, it is helpful to represent the formula not as a string but as a structured data format that captures the mathematical relationships and dependencies. This representation is achieved by parsing the LaTeX formulas into a symbolic expression format, essentially creating an operator tree. The



```
The derivative of a function $f$, referred to as $\frac{d}{dx}f(x)$, is defined
as the limit of $\frac{f(x+h) - f(x)}{h}$ as $h$ approaches $0$.
```

TEXT("The derivative of a function “, FUNC(f), “, referred to as “, DERIV(FUNC(f), VAR(x)), “, is defined as the limit of “, DIV(SUB(FUNC(f, ADD(VAR(x), VAR(h))), FUNC(f, VAR(x))), VAR(h)), “ as “, VAR(h), “ approaches “, INT(0), “.”)

VAR(x) → VAR(y), FUNC(f) → FUNC(g)

TEXT("The derivative of a function “, FUNC(g), “, referred to as “, DERIV(FUNC(g), VAR(y)), “, is defined as the limit of “, DIV(SUB(FUNC(g, ADD(VAR(y), VAR(h))), FUNC(g, VAR(y))), VAR(h)), “ as “, VAR(h), “ approaches “, INT(0), “.”)

```
The derivative of a function $g$, referred to as $g’(y)$, is defined
as the limit of $\left(g(h+y) - g(y)\right)/h$ as $h$ approaches $0$.
```

Figure 3: Visualization of the EquVG algorithm for a mathematical text.

symbolic representation categorizes elements into numbers, variables, functions, and other mathematical objects, establishing a structured relationship between them. This organization enables the identification and substitution of variables (`x`, `\alpha`, . . .) and generic functions (`f`, `g`, . . .) to derive a mathematically equivalent representation using different symbols. This substituted expression is then converted back into LaTeX format during the printing process, which includes the desired modifications of mathematical notation, such as writing $a \cdot a$ instead of a^2. In the following, we will provide a more detailed explanation of the three steps of EquVG: parsing, substituting, and printing.

LaTeX Parsing Parsing a LaTeX formula into a symbolic expression presents certain challenges. For instance, if a letter precedes parentheses, it can be interpreted either as a multiplication (with omitted multiplication symbol, e.g., $v(x + y) = v \cdot (x + y)$) or as a function call ($v(x + y)$). The symbol e could be either a variable or Euler’s number $e \approx 2.718$. Likewise, the symbol i might function as a variable (e.g., as a summation index in $\sum_{i=1}^{n} i^2$) or as the imaginary unit, sometimes expressed in LaTeX as i (`\mathrm{i}`) to avoid ambiguity. It is crucial for our purposes to determine whether a symbol is a substitutable variable or a constant. Otherwise, the imaginary unit might be incorrectly substituted by another symbol, resulting in a non-equivalent expression. We have addressed this issue, partially by applying heuristics that consider the context ($i = 1$ within the formula indicates a variable, while $i\pi$ indicates the imaginary unit, as in $e^{i\pi}$). Furthermore, we introduce a safeguard to handle cases where the parser is uncertain about whether to treat a symbol as a variable or the imaginary unit. In these cases, the symbol is represented in a way that prevents substitution while maintaining its appearance as the plain i, without enabling complex unit formatting options. Despite these measures, parsing can still fail in cases with unusual or malformed notation.

The formula parsing can be conceptually expanded to include the parsing of text containing LaTeX formulas. Such texts are referred to as *mathematical texts* in this study. The text parts remain unchanged during substitution and printing, only the formula parts are processed consistently by EquVG as shown in Figure 3. This allows to consistently change symbols in a mathematical text throughout all its formulas. Within a mathematical text, formulas are defined as text in between dollar signs (`$...$`), as used in LaTeX documents to write inline mathematical formulas.

Symbol Substitution The symbolic expression format allows the substitution of symbols by simply replacing all occurrences of a given symbol within the expression. The generation of a substitution (i.e., a mapping of symbols) involves two steps. Firstly, a subset of all symbols in the expression is randomly selected. Secondly, a new symbol is chosen for each selected symbol. The aim of the substitution process is to generate diverse formulas that are similar to formulas occurring in real-world scenarios. It is important to note that, intuitively, Eq. (6) with variables c and d appears more familiar for a binomial formula than Eq. (7), which uses Greek and uppercase Latin letters (λ and Z) that are not commonly used as variables in the context of binomial formulas. When variables are selected entirely at random from a uniform distribution over all Latin and Greek letters, unfamiliar symbol usage is more likely. This motivates the introduction of *symbol groups*, which categorizes similar variables or functions together. The defined symbol groups can be found in Table 10, along with a description of a typical mathematical context for each group. For instance, we have the group of indices $\{i, j, k, l\}$ and group of vectors $\{u, v, w\}$. It is worth noting that variables can belong to multiple groups. Given a symbol that should be substituted with a new symbol, all symbols from the relevant symbol group(s) are candidates. Additionally, the most common variable x is a candidate in every variable group to reflect its common usage. To add variety, a random symbol may be added by chance to the set of candidates, selected from commonly used lowercase or uppercase Latin letters or lowercase Greek letters. Symbols that refer to constants in certain contexts, such as e, i, and π, are excluded. Given the set of candidates, a random candidate is chosen. However, it is sometimes necessary (or at least useful) to make the symbol selection dependent on multiple substitution symbols. For instance, consider the Fundamental Theorem of Calculus: $\int_a^b f(x)\,dx = F(b) - F(a)$. Here, we have two generic functions, f and F. Whenever a symbol has related variants in the formula, such as uppercase and lowercase forms or corresponding Greek letters (e.g., a, A, α), the algorithm automatically preserves these relationships by restricting possible substitutions accordingly. For instance, substituting f and F by g and G, respectively, yields an equivalent version $\int_a^b g(x)\,dx = G(b) - G(a)$. Again, this is rather mathematically imprecise but aligns with the implicit assumptions on mathematical notation found in real-world datasets. Another possible generated version is $\int_{a_1}^{a_2} f(x)\,dx = F(a_2) - F(a_1)$ where a and b are substituted by indexed variables a_1 and a_2 respectively. In cases where multiple variables of the same variable group appear in the same formula, the generation algorithm may randomly perform such indexed substitutions. The indexing enforces the model not only to attend to the variable itself but also its modifications, in this case, its index.

Randomized LaTeX Printing In the final step of EquVG, the symbolic expression is converted back into LaTeX format. To further increase the variety of generated formulas, the LaTeX printer makes randomized printing decisions. These variations can be categorized into two main sources: mathematical and LaTeX notation. The parsed and printed formulas in Figure 3 are illustrating these differences. For example, the input text used the explicit notation for a derivative, `\frac{d}{dx}f(x)`, while the printed substituted expression uses the shorthand notation `g'(y)`. We developed a list of equivalent mathematical notations, where the printer randomly selects one of the available ones for printing. As another example in Figure 3, instead of the fraction notation with `\frac`, the printer used the forward slash `/` to denote division. Since addition is commutative, `h+y` is printed instead of `y+h`. These examples represent mathematical variations, as they express a mathematical concept in an equivalent way. In contrast, the usage of `\left` and `\right` commands represents LaTeX variations, since these commands are not essential for mathematical reasons but only for a differently rendered text. In addition to the already covered examples, the randomization of the LaTeX printer includes the notation of

- equalities (`x = y` vs. `y = x`),
- inequalities (`x > 0` vs. `0 < x`),
- multiplication symbols (`a \cdot b` vs. `a * b` vs. `a \times b` vs. `ab`),
- divisions (`2/n` vs. `2 \cdot n^{-1}` vs. `\frac{2}{n}` vs. `\frac2n`),
- integer powers (`a^3` vs. `a^2\cdot a` vs. `a \cdot a \cdot a`),
- inverse trigonometric functions (`\asin(x)` vs. `\arcsin(x)` vs. `\sin^{-1}(x)` vs. `(\sin(x))^{-1}`),
- higher order derivatives (`f'''(x)` vs. `f^{(3)}(x)` vs. `\frac{d^3}{dx^3} f(x)`),
- expected values (`\mathbb{E}[X]` vs. `\operatorname{E}[X]` vs. `E[X]`),
- matrix determinants (`\det(A)` vs. `|A|`),
- binomial coefficients (`\binom{n}{k}` vs. `{n \choose k}`),
- empty sets (`\emptyset` vs. `\varnothing` vs. `\{\}`), and
- natural logarithms (`\ln(x)` vs. `\log_e(x)`).

As real-world data uses different styles of notations, language models should be capable of understanding all commonly used notations. This is similar to the notion of synonyms in natural language. The randomized LaTeX printing provides an automation to diversify training data, such that models can learn the different notations. The combination of parsing, substituting, and printing results as part of EquVG is a powerful tool to increase the training data size significantly. Additionally, research has shown that using training data with substituted query-document pairs for MIR helps the model to less focus on shallow features such as variable overlapping (Reusch et al., 2024), confirming the usage of substitutions in EquVG.

5.2 FalseVG: Generating Challenging Negative Examples

We believe that a classification task determining whether two formulas describe the same mathematical concept helps the model to encode mathematics more effectively. To train models on such a task, we require both positive and negative formula pairs, similar to a MIR training. While positive pairs are often readily available in datasets, identifying meaningful negative pairs is more challenging, as datasets rarely contain explicit negative examples.

A common approach to extract negative pairs is random sampling from datasets by pairing two random formulas. However, this may lead to simplified feature extractions. For instance, the model might learn to simply check for the presence of an important function, like whether both formulas contain the determinant function `\det`. Given a random negative document, a naive classifier that checks if a determinant is part of the formula would likely perform well, due to the rarity of the determinant function across most mathematical contexts. The language model may adapt to this behavior during training.

To prevent models from learning such easy shortcuts instead of the true semantic understanding, researchers have successfully used challenging negative examples in other domains (Cai & Liu, 2020; Qiu et al., 2021). We introduce the Falsified Version Generation (FalseVG) algorithm to generate *falsified versions* of a given formula, meaning a similar-looking but *not* mathematically equivalent formula. Since the formulas are already parsed into a symbolic expression for EquVG, we can simply use and modify this representation. We have developed and implemented eight modification strategies, which are described below. Table 9 provides illustrative examples of each strategy. Similar to EquVG, these strategies can also be applied to mathematical texts, by applying the strategies to the text's formulas.

Equality Falsifying an *equality* can be achieved by inserting or removing a term on one side of the equality. This can be done either at the outermost level (e.g., changing $\sin(x) = \ldots$ to $\sin(x) + 1 = \ldots$) or within a sub-expression (e.g., changing $\sin(x) = \ldots$ to $\sin(x+1) = \ldots$). When inserting a term, the algorithm selects either a subexpression from the entire formula, a random new variable, or a random number. Importantly, the algorithm avoids modifications that will not change the validity of an equality, such as adding zero or multiplying by one. This strategy enforces the model to focus on the entire formula and long-term dependencies.

Inequality To falsify an *inequality*, we simply invert the inequality symbol. Thus, the symbol $\leq$ is replaced by $>$ and vice versa. The same replacement holds for $\geq$ and $<$. The not equals symbol $\neq$ can be replaced by $=$, but not vice versa (as $=$ indicates an equality). Similarly to the strategy equality, the model is forced to encode long-term dependencies using this strategy.

Swap The strategy *swap* involves altering unary and binary functions. Unary functions such as sine, square root, or logarithm get replaced by different random unary functions. In the case of binary non-commutative functions, we swap the order of the two arguments. These non-commutative functions are subtraction, division, and exponentiation (e.g., x^2 becomes 2^x). These changes enforce the model to rely on the order of operands rather than just token occurrences in a random order.

Variable The strategy *variable* essentially aims to split a single variable (e.g., a in $(\boldsymbol{a}+b)^2 = \boldsymbol{a}^2+2\boldsymbol{a}b = b^2$) into two (e.g., into a and c in $(\boldsymbol{c}+b)^2 = \boldsymbol{a}^2 + 2\boldsymbol{c}b + b^2$). Specifically, if a variable occurs at least twice in the formula, it might be randomly replaced by another variable for a proper and nonempty subset of its occurrences (i.e., at least one occurrence is replaced and at least one occurrence remains unchanged). This strategy enforces the model to check for a consistent use of symbols in the entire formula.

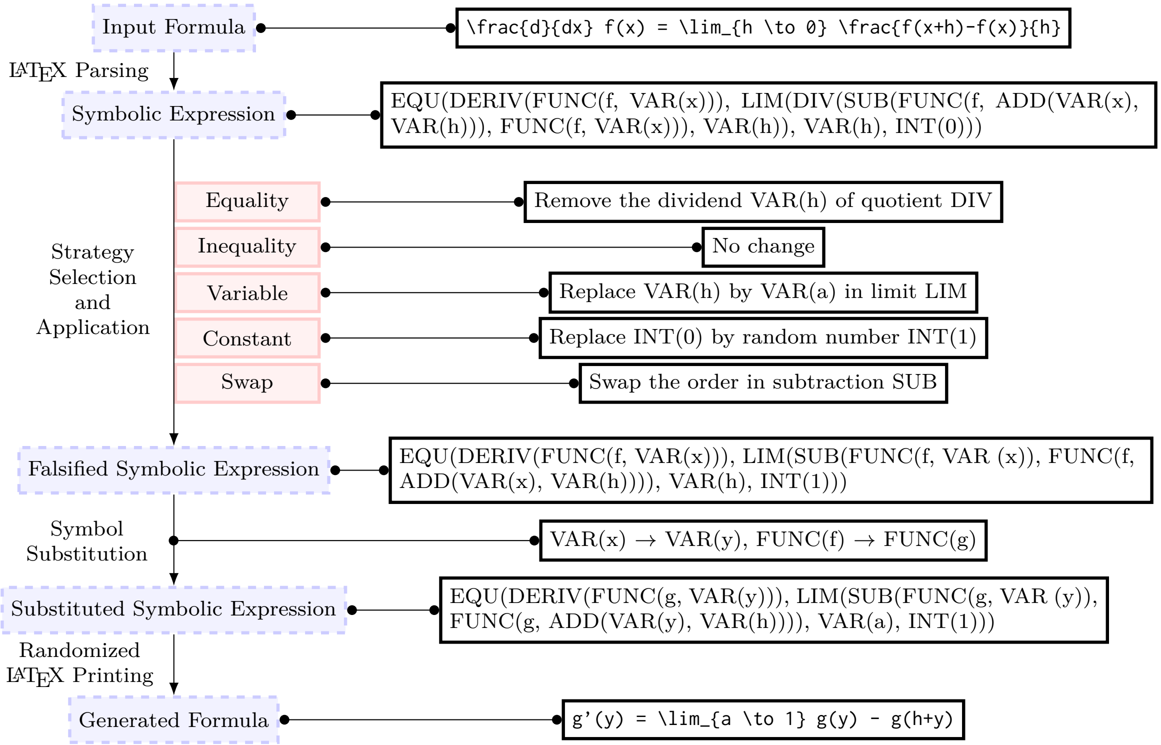

Input Formula
\frac{d}{dx} f(x) = \lim_{h \to 0} \frac{f(x+h)-f(x)}{h}
LaTeX Parsing
Symbolic Expression
EQU(DERIV(FUNC(f, VAR(x))), LIM(DIV(SUB(FUNC(f, ADD(VAR(x), VAR(h))), FUNC(f, VAR(x))), VAR(h)), VAR(h), INT(0)))
Strategy Selection and Application
Equality
Remove the dividend VAR(h) of quotient DIV
Inequality
No change
Variable
Replace VAR(h) by VAR(a) in limit LIM
Constant
Replace INT(0) by random number INT(1)
Swap
Swap the order in subtraction SUB
Falsified Symbolic Expression
EQU(DERIV(FUNC(f, VAR(x))), LIM(SUB(FUNC(f, VAR (x)), FUNC(f, ADD(VAR(x), VAR(h)))), VAR(h), INT(1)))
Symbol Substitution
VAR(x) → VAR(y), FUNC(f) → FUNC(g)
Substituted Symbolic Expression
EQU(DERIV(FUNC(g, VAR(y))), LIM(SUB(FUNC(g, VAR (y)), FUNC(g, ADD(VAR(y), VAR(h)))), VAR(a), INT(1)))
Randomized LaTeX Printing
Generated Formula
g'(y) = \lim_{a \to 1} g(y) - g(h+y)

Figure 4: Visualization of the FalseVG algorithm.

Constant The strategy *constant* focuses on numbers (1, 2, e, π, ∞, ...) as well as variables that are typically considered to be constant within an expression, such as the upper limit n of an indexed sum like $\sum_{i=1}^{n} i^2$. These constants are replaced by other constants enforcing the model to learn what tokens a certain formula should contain.

Distribute The strategy *distribute* is inspired by the distributive law, a fundamental mathematical rule relating two binary functions. A standard example for real numbers is that multiplication distributes over addition since $x \cdot (y+z) = x \cdot y + x \cdot z$ holds for all real numbers x, y, z. This rule motivates this strategy, which applies a modified distributive law to non-distributive functions. Specifically, for a unary function f and a binary function $\oplus$ in infix notation, the relation $f(x \oplus y) = f(x) \oplus f(y)$ is (falsely) assumed. We use addition and multiplication as binary functions and the logarithm, factorial, power with fixed base, and trigonometric functions for the unary function. This readily results in examples where commonly known identities are falsified, e.g., the falsified product of powers rule is $2^x \cdot 2^y = 2^{x \cdot y}$ (instead of the correct $2^x \cdot 2^y = 2^{x+y}$). The falsified sine additivity yields $\sin(x+y) = \sin(x) + \sin(y)$ (instead of $\sin(x+y) = \sin(x)\cos(y) + \cos(x)\sin(y)$). This strategy enforces the model to notice the presence of parantheses and to enhance its understanding of operator relationships, including precedence.

Manual While all previous strategies focused on modifying a formula by applying generally valid transformation rules to falsify it, this strategy relies on *manual* transformation or replacement rules. These rules refer to the specifically newly created NMFT dataset (see Section 4). The rules can be explicitly given falsified versions (e.g., $\forall n \in \mathbb{N} : n! = 1 \cdot 2 \cdot n$), references to different but similar formulas (e.g., law of cosines for the Pythagorean theorem), or falsifying replacement rules. For example, a formula replacement might change $\forall n \in \mathbb{N}$ to $\forall n \in \mathbb{R}$ if the quantified term only holds for natural but not for real numbers. These rules are also applicable to mathematical texts, where, for instance, *natural* can be replaced by *real*.

Random The simplest approach to generate a falsified formula is to use a *random* formula, meaning an earlier generated equivalent version of a different formula. This approach is especially important to increase the models' robustness in real-world applications, where most of the input pairs are not inherently challenging.

Name	**Hugging Face Identifier**	**Original Dataset(s)**	**Raw Entries**	**Generated Versions**	**⌀ v.p.f.**	**Max v.p.f.**
MF	anonymous	AMPS	30,985	958,735	30.9	101
		ARQMath	55,894	2,257,826	43.3	101
		Both	82,765	3,198,108	38.6	101
MT	anonymous	AMPS	62,099	2,542,015	40.9	101
		ARQMath	690,333	4,480,369	6.5	96
		Both	752,428	7,022,384	9.3	101
NMF	anonymous	NMFT	71/ 522	23,707,392	333,906	400,000
MFR	anonymous	NMFT	71/ 522	23,702,560	334,092	400,000

Table 5: Summary of the generated datasets. The abbreviation *v.p.f.* stands for *versions per formula*. For MF, the *Generated Entries* values do not sum up from AMPS and Answer Retrieval for Questions on Math (ARQMath) to *Both* due to duplicate removal. The raw values of NMF and MFR refer to the number of mathematical identities and the total number of provided version templates of these identities, respectively.

The complete FalseVG algorithm is summarized in Figure 4. It involves applying a random subset of the strategies to a parsed symbolic expression. Note that some strategies are not applicable to certain formulas, resulting in no changes. However, if at least one strategy succeeds, a falsified symbolic expression is generated. Finally, a random symbol substitution and randomized LaTeX printing are performed to create the final formula as a string, identically to EquVG.

6 Generated Datasets

This section presents four datasets generated using MAMUT employing EquVG and FalseVG. Our implementation, built on SymPy (Meurer et al., 2017), is detailed in Appendix B.1. Table 5 summarizes key statistics of the generated datasets, including Hugging Face identifiers, while Appendix B.2 reports example entries. All entries ensure uniqueness at the string level. Our dataset generation code is publicly available[1].

Two of the generated datasets (NMF and MFR) are based on the specifically created NMFT dataset (see Section 4), while the other two datasets (MF and MT) are derived from two existing diverse sources that combine natural language with mathematical notation: ARQMath (Mansouri et al., 2022a) and the Khan Academy problems in AMPS (Hendrycks et al., 2021). While we focus on these sources, MAMUT is applicable to any mathematical corpus containing LaTeX notation. ARQMath, sourced from the Mathematics Stack Exchange, benefits from a user-rating system that ensures high-quality discussions and problem-solving content. The Khan Academy problems in AMPS provide structured exercises used for educational purposes. Example dataset entries are shown in Appendix A.

Mathematical Formulas (MF) This dataset consists exclusively of mathematical formulas extracted from AMPS and ARQMath, enriched with variations by the EquVG algorithm. However, not all formulas from these raw datasets are included in the MF dataset. Only formulas are selected being suitable for a Masked Language Modeling (MLM) task (Devlin et al., 2019), where a masked token's value can be concluded by the remaining context of the formula. For example, a masked formula such as $\pi >$ `[MASK]` has infinite algebraic solutions, like 3, 0, or any other value that can fill the masked position in a mathematical valid sense. Therefore, the formulas are restricted to equalities and implications to ensure meaningful inferences. Additionally, the (general) validity of each used expression is verified using SymPy to ensure high data quality. In case of an equation without general validity (e.g., $x^2 = 2$) but with existing solution(s) found by SymPy, the equation can be transformed into an implication (e.g., $x^2 = 2 \Rightarrow x = -\sqrt{2}$ or $x = \sqrt{2}$). A few

---

[1] anonymous

examples of extracted formulas are (original LaTeX formatting is preserved):

$$\tan(x) = \sin(x)/\cos(x),$$
$$-\frac{2}{5} \div -\frac{1}{6} = -\frac{2}{5} \times -\frac{6}{1},$$
$$3x = 210 \Rightarrow x = 70,$$
$$\frac{3}{13} - \frac{2}{13} = \frac{1}{13},$$
$$e^{2\pi i} = (e^{\pi i})^2 = (-1)^2 = 1,$$
$$\sqrt{25} = 5,$$
$$(n+1) \times (n-1)! = \frac{(n+1) \times n \times (n-1)!}{n} = \frac{(n+1)!}{n}.$$

Mathematical Texts (MT) While the previous dataset MF focuses exclusively on mathematical formulas, MT focuses on the relationship between mathematical formulas and natural language. Similarly to MF, MT is generated using the AMPS and ARQMath datasets, along with applying EquVG, which consistently changes variable names across the text and prints the LaTeX formulas in different ways. We only consider texts containing at least five formulas. Questions and answers of ARQMath are treated as separate text, while the AMPS data is treated as a single text where question and hints are concatenated. We generate up to 100 additional versions for each suitable input.

Named Mathematical Formulas (NMF) This dataset associates the name of a mathematical identity with either its formula or a describing text. It is derived from NMFT by applying both, EquVG and FalseVG, resulting in diverse positive and negative pairs. This data could be used to train a classifier that predicts whether a formula is a valid representation of an identity's name, using a Next Sentence Prediction (NSP)-like task (Devlin et al., 2019). In a typical NSP task, each positive pair is matched with a random negative pair, which changes when the positive pair is reused. To enhance training, we create an imbalanced dataset with four times more negative than positive pairs. This allows for training where positive pairs remain unchanged across epochs, while negative pairs vary between epochs (and remain challenging). With a maximum of four epochs, the model encounters unique negative pairs in each iteration. NMF originates from 71 mathematical identities, each with multiple base versions used to generate up to 400k versions per identity. About 60% of the NMF entries are textual descriptions, and the rest are pure mathematical formulas. For 20 of the 71 mathematical identities, fewer than 400k versions exist, as they offer fewer possibilities for generating versions, such as limited substitution options or fewer opportunities for creating randomized LaTeX.

Mathematical Formula Retrieval (MFR) This dataset consists of formula pairs, classified as either mathematical equivalent or not. It is constructed by pairing each true formula version from NMF with an equivalent version and four falsified versions of that identity, all randomly sourced from NMF. This approach preserves the positive-to-negative pair ratio while ensuring that negative pairs remain challenging. MFR can be used to train a MIR system for querying relevant formulas based on a similar formula, like a NSP task.

7 Conclusion

Mathematical formulas are essential to communicate complex and abstract concepts in various scientific fields. To effectively encode the unique structure of mathematical language, specialized mathematical language models are required. We developed MAMUT, a framework based on SymPy (Meurer et al., 2017) that generates equivalent and falsified versions of LaTeX formulas through parsing, substituting, possibly falsifying, and printing again into LaTeX format. MAMUT diversifies and expands datasets, as demonstrated by four generated large, high-quality datasets: MF, MT, NMF and MFR, all publicly available on Hugging Face[2] (see Table 5). These datasets can be leveraged for further mathematical pre-training of language models utilizing tasks such as Masked Language Modeling (MLM) and Causal Language Modeling (CLM) to predict equation parts, an Next Sentence Prediction (NSP) variant that predicts if equations are equivalent, or contrastive learning between positive and negative samples to learn equation embeddings.

---

[2] anonymous

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Akiko Aizawa and Michael Kohlhase. *Mathematical Information Retrieval*, pp. 169–185. Springer Singapore, Singapore, 2021. ISBN 978-981-15-5554-1. doi: 10.1007/978-981-15-5554-1_12. URL `https://doi.org/10.1007/978-981-15-5554-1_12`.

Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A pretrained language model for scientific text. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3615–3620, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1371. URL `https://aclanthology.org/D19-1371/`.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Wenjie Cai and Qiong Liu. Image captioning with semantic-enhanced features and extremely hard negative examples. *Neurocomputing*, 413:31–40, 2020. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.06.112. URL `https://www.sciencedirect.com/science/article/pii/S0925231220311012`.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168.

Pankaj Dadure, Partha Pakray, and Sivaji Bandyopadhyay. Mathematical information retrieval: A review. *ACM Comput. Surv.*, 57(3), November 2024. ISSN 0360-0300. doi: 10.1145/3699953. URL `https://doi.org/10.1145/3699953`.

Xuan-Quy Dao and Ngoc-Bich Le. Investigating the effectiveness of ChatGPT in mathematical reasoning and problem solving: Evidence from the vietnamese national high school graduation examination. *arXiv preprint arXiv:2306.06331*, abs/2306.06331, 2023. URL `https://arxiv.org/abs/2306.06331`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of the Association for Computational Linguistics*, 2019. URL `https://api.semanticscholar.org/CorpusID:52967399`.

Jingxuan Fan, Sarah Martinson, Erik Y Wang, Kaylie Hausknecht, Jonah Brenner, Danxian Liu, Nianli Peng, Corey Wang, and Michael P Brenner. HARDMath: A benchmark dataset for challenging problems in applied mathematics. *arXiv preprint arXiv:2410.09988*, 2024.

Meng Fang, Xiangpeng Wan, Fei Lu, Fei Xing, and Kai Zou. Mathodyssey: Benchmarking mathematical problem-solving skills in large language models using odyssey math data. *arXiv preprint arXiv:2406.18321*, 2024.

Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, and Ming Zhou. CodeBERT: A pre-trained model for programming and natural languages. In Trevor Cohn, Yulan He, and Yang Liu (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 1536–1547, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.139. URL `https://aclanthology.org/2020.findings-emnlp.139/`.

Zheng Gong, Kun Zhou, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. Continual pre-training of language models for math problem understanding with syntax-aware memory network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5923–5933, 2022.

André Greiner-Petter, Abdou Youssef, Terry Ruas, Bruce R Miller, Moritz Schubotz, Akiko Aizawa, and Bela Gipp. Math-word embedding in math search and semantic extraction. *Scientometrics*, 125:3017–3046, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Bat-Sheva Ilany, Bruria Margolin, et al. Language and mathematics: Bridging between natural language and mathematical language in solving problems in mathematics. *Creative Education*, 1(03):138, 2010.

Shinil Kim, Seon Yang, and Youngjoong Ko. Mathematical equation retrieval using plain words as a query. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 2407–2410, 2012.

Giovanni Yoko Kristianto, Goran Topic, and Akiko Aizawa. MCAT math retrieval system for NTCIR-12 MathIR Task. In *NTCIR*, 2016.

Zenan Li, Zhi Zhou, Yuan Yao, Yu-Feng Li, Chun Cao, Fan Yang, Xian Zhang, and Xiaoxing Ma. Neuro-symbolic data generation for math reasoning, 2024. URL `https://arxiv.org/abs/2412.04857`.

Wentao Liu, Hanglei Hu, Jie Zhou, Yuyang Ding, Junsong Li, Jiayi Zeng, Mengliang He, Qin Chen, Bo Jiang, Aimin Zhou, et al. Mathematical language models: A survey. *arXiv preprint arXiv:2312.07622*, 2023.

Zimu Lu, Aojun Zhou, Houxing Ren, Ke Wang, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. Mathgenie: Generating synthetic data with question back-translation for enhancing mathematical reasoning of llms. *arXiv preprint arXiv:2402.16352*, 2024.

Behrooz Mansouri, Shaurya Rohatgi, Douglas W Oard, Jian Wu, C Lee Giles, and Richard Zanibbi. Tangent-CFT: An embedding model for mathematical formulas. In *Proceedings of the 2019 ACM SIGIR international conference on theory of information retrieval*, pp. 11–18, 2019.

Behrooz Mansouri, Vít Novotnỳ, Anurag Agarwal, Douglas W Oard, and Richard Zanibbi. Overview of arqmath-3 (2022): Third clef lab on answer retrieval for questions on math. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pp. 286–310. Springer, 2022a.

Behrooz Mansouri, Douglas W. Oard, and Richard Zanibbi. Contextualized formula search using math abstract meaning representation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, pp. 4329–4333, New York, NY, USA, 2022b. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557567. URL `https://doi.org/10.1145/3511808.3557567`.

Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. SymPy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, January 2017. ISSN 2376-5992. doi: 10.7717/peerj-cs.103. URL `https://doi.org/10.7717/peerj-cs.103`.

Vít Novotný and Michal Štefánik. Combining sparse and dense information retrieval. In Guglielmo Faggioli, Nicola Ferro, Allan Hanbury, and Martin Potthast (eds.), *Proceedings of the Working Notes of CLEF 2022*, pp. 104–118. CEUR-WS, 2022. URL `http://ceur-ws.org/Vol-3180/paper-06.pdf`.

Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. Mathbert: A pre-trained model for mathematical formula understanding. *ArXiv*, abs/2105.00377, 2021. URL `https://arxiv.org/abs/2105.00377`.

Felix Petersen, Moritz Schubotz, Andre Greiner-Petter, and Bela Gipp. Neural machine translation for mathematical formulae, 2023. URL `https://arxiv.org/abs/2305.16433`.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*, 2024.

Yao Qiu, Jinchao Zhang, Huiying Ren, and Jie Zhou. Challenging instances are worth learning: Generating valuable negative samples for response selection training. *arXiv preprint arXiv:2109.06538*, 2021. URL `https://arxiv.org/abs/2109.06538`.

Anja Reusch, Maik Thiele, and Wolfgang Lehner. Transformer-encoder and decoder models for questions on math. In *Conference and Labs of the Evaluation Forum*, 2022. URL `https://ceur-ws.org/Vol-3180/paper-07.pdf`.

Anja Reusch, Julius Gonsior, Claudio Hartmann, and Wolfgang Lehner. Investigating the usage of formulae in mathematical answer retrieval. In *European Conference on Information Retrieval*, pp. 247–261. Springer, 2024.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. *arXiv preprint arXiv:1904.01557*, 2019.

Alexander Scarlatos and Andrew Lan. Tree-based representation and generation of natural and mathematical language. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3714–3730, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.205. URL `https://aclanthology.org/2023.acl-long.205/`.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Ruoqi Shen, Sébastien Bubeck, Ronen Eldan, Yin Tat Lee, Yuanzhi Li, and Yi Zhang. Positional description matters for transformers arithmetic, 2023. URL `https://arxiv.org/abs/2311.14737`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL `https://arxiv.org/abs/2302.13971`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Weihao You, Shuo Yin, Xudong Zhao, Zhilong Ji, Guoqiang Zhong, and Jinfeng Bai. Mumath: Multi-perspective data augmentation for mathematical reasoning in large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2932–2958, 2024.

Richard Zanibbi, Akiko Aizawa, Michael Kohlhase, Iadh Ounis, Goran Topic, and Kenny Davila. Ntcir-12 mathir task overview. In *NTCIR*, 2016.

Richard Zanibbi, Douglas W Oard, Anurag Agarwal, and Behrooz Mansouri. Overview of ARQMath 2020: CLEF lab on answer retrieval for questions on math. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings 11*, pp. 169–193. Springer, 2020.

Richard Zanibbi, Behrooz Mansouri, Anurag Agarwal, et al. Mathematical information retrieval: Search and question answering. *Foundations and Trends® in Information Retrieval*, 19(1-2):1–190, 2025.

Bo-Wen Zhang, Yan Yan, Lin Li, and Guang Liu. Infinity <scp>math:</scp> a scalable instruction tuning dataset in programmatic mathematical reasoning. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, CIKM '24, pp. 5405–5409. ACM, October 2024. doi: 10.1145/3627673.3679122. URL `http://dx.doi.org/10.1145/3627673.3679122`.

Wei Zhong, Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. One blade for one purpose: advancing math information retrieval using hybrid search. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 141–151, 2023.

A Original Datasets

In Table 6, we present one version of each mathematical identity of NMFT, while this entire raw dataset is available on Hugging Face[3] as part of the NMF dataset files. Subsequently, Table 7 provides an example entry of ARQMath (Mansouri et al., 2022a) from the Mathematical Stack Exchange, while Table 8 shows an example of Auxiliary Mathematics Problems and Solutions (AMPS) (Hendrycks et al., 2021).

Name	**Formula**
Addition Theorem for Cosine	$\forall\alpha,\beta\in\mathbb{R}:\cos(\alpha+\beta)=\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)$
Addition Theorem for Cosine	$\forall\alpha,\beta\in\mathbb{R}:\cos(\alpha+\beta)=\cos(\alpha)\cos(\beta)-\sin(\alpha)\sin(\beta)$
Addition Theorem for Sine	$\forall\alpha,\beta\in\mathbb{R}:\sin(\alpha+\beta)=\sin(\alpha)\cos(\beta)+\cos(\alpha)\sin(\beta)$
Addition Theorem for Tangent	$\forall\alpha,\beta\in\mathbb{R}:\tan(\alpha+\beta)=\frac{\tan(\alpha)+\tan(\beta)}{1-\tan(\alpha)\tan(\beta)}$
Alternating Harmonic Series	$\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{n}=1-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}+\ldots=\ln(2)$
Basel Problem	$\sum_{n=1}^{\infty}\frac{1}{n^2}=\frac{1}{1^2}+\frac{1}{2^2}+\frac{1}{3^2}+\frac{1}{4^2}+\frac{1}{5^2}+\frac{1}{6^2}+\ldots=\frac{\pi^2}{6}$
Bayes' Theorem	$\mathbb{P}(A\|B)=\frac{\mathbb{P}(B\|A)\cdot\mathbb{P}(A))}{\mathbb{P}(B)}$
Bernouilli Inequality	$\forall x\geq-1,\forall\alpha>1\Rightarrow(1+x)^{\alpha}\geq1$
Binomial Coefficient Formula	$\forall n,k\in\mathbb{N},n\geq k:\binom{n}{k}=\frac{n!}{k!(n-k)!}$
Binomial Distribution	$\mathbb{P}(X=k)=\binom{n}{k}p^k\cdot(1-p)^{n-k}$
Binomial Series	$\forall\alpha,x\in\mathbb{C}>0:\|x\|<1\Rightarrow(1+x)^{\alpha}=\sum_{k=0}^{\infty}\binom{\alpha}{k}x^k$
Binomial Theorem	$\forall a,b\in\mathbb{R}\forall n\in\mathbb{N}:(a+b)^n=\sum_{k=0}^{n}\binom{n}{k}a^{n-k}b^k$
Chain Rule	$\frac{d}{dx}\left[f(g(x))\right]=f'(g(x))\cdot g'(x)$
Complex Number Division	$\forall a,b,c,d\in\mathbb{R}:\frac{a+b\mathrm{i}}{c+d\mathrm{i}}=\frac{(ac+bd)+(bc-ad)i}{c^2+d^2}$
Complex Number Inverse	$\forall z\in\mathbb{C}:z=a+b\mathrm{i}\Rightarrow z^{-1}=\frac{a}{a^2+b^2}-\frac{b}{a^2+b^2}\mathrm{i}$
Complex Number Multiplication	$\forall a,b,c,d\in\mathbb{R}:(a+b\mathrm{i})\cdot(c+d\mathrm{i})=(ac-bd)+(ad+bc)\mathrm{i}$
Complex Number Sum	$\forall a,b,c,d\in\mathbb{R}:(a+b\mathrm{i})+(c+d\mathrm{i})=(a+c)+(b+d)\mathrm{i}$
Cosine Function Definition	$\forall x\in\mathbb{R}:\cos(x)=\sum_{n=0}^{\infty}\frac{(-1)^n}{(2n)!}x^{2n}$
Covariance	$\mathrm{Cov}[X,Y]=\mathrm{E}[(X-\mathrm{E}[X])(Y-\mathrm{E}[Y])]$
De Morgan Law	$\forall x,y:\neg(x\wedge y)=\neg x\vee\neg y$

Table 6: The 71 mathematical identities of the NMFT dataset.

---

[3] anonymous

(Continued from previous page)

Name	**Formula**
Derivative of Inverse Function	$\frac{d}{dx}\left[f^{-1}(x)\right] = \frac{1}{f'(f^{-1}(x))}$
Derivative of a Function	$f'(x) = \lim_{h\to 0} \frac{f(x+h)-f(x)}{h}$
Determinant of 2x2 Matrix	$\det\left(\begin{smallmatrix} a & b \\ c & e \end{smallmatrix}\right) = a \cdot e - b \cdot c$
Determinant of 3x3 Matrix	$\det\begin{pmatrix} a & b & c \\ d & e & f \\ g & h & j \end{pmatrix} = a \cdot \det\begin{pmatrix} e & f \\ h & j \end{pmatrix} - b \cdot \det\begin{pmatrix} d & f \\ g & j \end{pmatrix} + c \cdot \det\left(\begin{smallmatrix} d & e \\ g & h \end{smallmatrix}\right)$
Distributive Law of Sets	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
Euler's Formula	$\forall \alpha \in \mathbb{C} : e^{\mathrm{i}\alpha} = \cos(\alpha) + \mathrm{i}\sin(\alpha)$
Euler's Formula for Polyhedra	$V - E + F = 2$
Euler's Identity	$e^{\mathrm{i}\pi} + 1 = 0$
Euler's Number	$e = \lim_{n\to\infty}(1 + 1/n)^n$
Expected Value	$\mathbb{E}(X) = \sum_{i=1}^{n} x_i \mathbb{P}(X = x_i)$
Exponential Function	$\forall x \in \mathbb{R} : \lim_{n\to\infty} (1 + x/n)^n = e^x$
Factorial	$\forall n \in \mathbb{N} : n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \ldots \cdot n$
First Binomial Formula	$\forall a, b \in \mathbb{R} : (a + b)^2 = a^2 + 2ab + b^2$
Fundamental Theorem of Calculus	$\int_a^b f(x)\, dx = F(b) - F(a)$
Gamma Function	$\forall n \in \mathbb{N} : \Gamma(n) = \int_0^\infty x^{n-1} e^{-x} dx = (n-1)!$
Gaussian Integral	$\int_{-\infty}^{\infty} exp(-x^2)dx = \sqrt{\pi}$
Geometric Series	$\sum_{n=0}^{\infty} r^n = \frac{1}{1-r}$
Gregory-Leibniz Series	$\sum_{n=0}^{\infty}(-1)^n \cdot \frac{1}{2n+1} = \frac{\pi}{4}$
Harmonic Series	$\sum_{n=1}^{\infty} \frac{1}{n} = \infty$
Hölder Inequality	$\forall p, q > 1, \frac{1}{p} + \frac{1}{q} = 1, \forall x, y \in \mathbb{R}^n$ $\Rightarrow \sum_{i=1}^{n} \lvert x_i y_i \rvert \leq \left(\sum_{i=1}^{n} \lvert x_i \rvert^p\right)^{\frac{1}{p}} \cdot \left(\sum_{i=1}^{n} \lvert y_i \rvert^q\right)^{\frac{1}{q}}$
Integration by Parts	$\int f(x)g'(x)\, dx = f(x)g(x) - \int g(x)f'(x)\, dx$
Inverse of 2x2 Matrix	$\left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right)^{-1} = \frac{1}{ad-bc}\left(\begin{smallmatrix} d & -b \\ -c & a \end{smallmatrix}\right)$
Law of Cosines	$c^2 = a^2 + b^2 - 2ab\cos(C)$
Law of Large Numbers	$\lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} x_i = [E](X)$
Law of Sines	$\frac{\sin(A)}{a} = \frac{\sin(B)}{b} = \frac{\sin(C)}{c}$
Law of Total Probability	$\mathbb{P}(A) = \sum_{i=1}^{n} \mathbb{P}(A \vert B_i)\mathbb{P}(B_i)$
Logarithm Power Rule	$\forall b \in \mathbb{R}, b > 0, b \neq 1, \forall x, r \in \mathbb{R}, x > 0 : \log_b(x^r) = r \cdot \log_b(x)$
Logarithm Product Rule	$\forall b \in \mathbb{R}, b > 0, b \neq 1, \forall x, y > 0 : \log_b(xy) = \log_b(x) + \log_b(y)$
Logarithm Quotient Rule	$\forall b \in \mathbb{R}, b > 0, b \neq 1, \forall x, y > 0 : \log_b(x/y) = \log_b(x) - \log_b(y)$
Minkowski Inequality	$\forall p > 1 \Rightarrow \sum_{i=1}^{n} \lvert x_i + y_i \rvert^{\frac{1}{p}} \leq \left(\sum_{i=1}^{n} \lvert x_i \rvert^p\right)^{\frac{1}{p}} + \left(\sum_{i=1}^{n} \lvert y_i \rvert^p\right)^{\frac{1}{p}}$
Multiplication of 2x2 Matrix	$A = \left(\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right), B = \begin{pmatrix} e & f \\ g & h \end{pmatrix} \Rightarrow A \cdot B = \begin{pmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{pmatrix}$
Normal Distribution	$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/(2\sigma^2)}$

Table 6: The 71 mathematical identities of the NMFT dataset.

(Continued from previous page)

Name	**Formula**
Pascal's Rule	$\forall n, k \in \mathbb{N}: \binom{n+1}{k+1} = \binom{n}{k+1} + \binom{n}{k}$
Poisson Distribution	$\mathbb{P}(X = k) = \frac{e^{-\lambda}\lambda^k}{k!}$
Power Rule	$\forall n \in \mathbb{R}, n \neq 0: \frac{d}{dx}\left(x^n\right) = nx^{n-1}$
Principle of Inclusion-Exclusion	$\|A \cup B\| = \|A\| + \|B\| - \|A \cap B\|$
Product Rule	$\frac{d}{dx}[u(x) \cdot v(x)] = u'(x) \cdot v(x) + u(x) \cdot v'(x)$
Pythagorean Identity	$\forall \alpha \in \mathbb{R}: \sin^2(\alpha) + \cos^2(\alpha) = 1$
Pythagorean Theorem	$a^2 + b^2 = c^2$
Quadratic Formula	$\forall a, b, c \in \mathbb{R}, a \neq 0: a \cdot x^2 + b \cdot x + c = 0 \Rightarrow x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
Quotient Rule	$\forall b \in \mathbb{R}, b > 0, b \neq 1, \forall x, y > 0: \log_b(x/y) = \log_b(x) - \log_b(y)$
Riemann Zeta Function	$\forall z \in \mathbb{C}, \mathrm{Re}(z) > 1: \zeta(z) = \sum_{n=1}^{\infty} \frac{1}{n^z}$
Rule de l'Hôpital	$\lim_{x \to a} \frac{f(x)}{g(x)} = \lim_{x \to a} \frac{f'(x)}{g'(x)}$
Second Binomial Formula	$\forall a, b \in \mathbb{R}: (a - b)^2 = a^2 - 2a \cdot b + b^2$
Sine Function Definition	$\forall x \in \mathbb{R}: \sin(x) = \sum_{n=0}^{\infty} (-1)^n/(2n+1)!x^{2n+1}$
Stirling Approximation	$\forall n \in \mathbb{N}: n! \approx \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$
Taylor Series	$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x - a)^n$
Third Binomial Formula	$\forall a, b \in \mathbb{R}: (a + b)(a - b) = a^2 - b^2$
Variance	$\mathbb{V}\mathrm{ar}[X] = \mathrm{E}\left[(X - \mathrm{E}[X])^2\right]$
Wallis Product	$\prod_{n=1}^{\infty} \frac{4n^2}{4n^2 - 1} = \frac{\pi}{2}$
Young Inequality	$\forall p, q > 1, 1/p + 1/q = 1, \forall a, b \geq 0 \Rightarrow ab \leq \frac{a^p}{p} + \frac{b^q}{q}$
pq Formula	$\forall p, q \in \mathbb{R}: x^2 + px + q = 0 \Rightarrow x_{1,2} = -\frac{p}{2} \pm \sqrt{\frac{p^2}{4} - q}$

Table 6: The 71 mathematical identities of the NMFT dataset.

B MAMUT

Table 9 shows examples of the strategies for generating falsified formulas of FalseVG. Table 10 reports the used symbol groups for the symbol substitution of EquVG.

B.1 Implementation

As discussed in Section 6, the MAMUT relies on the EquVG and FalseVG algorithms, which generate equivalent or falsified versions of a given formula. We implemented these algorithms using the Python library SymPy 1.12, which is an open-source symbolic mathematics library with computer algebra system features (Meurer et al., 2017). This library includes a LaTeX parser for converting expressions into an internal SymPy representation, which can then be printed again into the LaTeX format. The SymPy formula representation is a symbolic expression, as required for EquVG and FalseVG, and supports the substitution of variables and generic functions. However, the built-in SymPy parser had limitations in handling various mathematical notations. The parsing capability has been expanded during this work, including the parsing of matrices, sets, derivatives, and various operators ($\pm$, $\cup$, $\cap$, $\mathbb{E}[X]$, $\mathbb{V}\mathrm{ar}[X]$, $\ldots$). Additionally, the SymPy LaTeX parsing was expanded to support a wider range of mathematical expressions through the implementation of an adaptive hybrid approach. This approach introduces a SymPy-like expression that enables safe

Title	Derivative of sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$
Question	In my AI textbook there is this paragraph, without any explanation. The sigmoid function is defined as follows: “$\sigma(x) = \frac{1}{1+e^{-x}}$. This function is easy to differentiate because $\frac{d\sigma(x)}{d(x)} = \sigma(x) \cdot (1 - \sigma(x))$.“ It has been a long time since I've taken differential equations, so could anyone tell me how they got from the first equation to the second?
Answer 1	Consider $f(x) = \dfrac{1}{\sigma(x)} = 1 + e^{-x}$. Then, on the one hand, the chain rule gives $f'(x) = \frac{d}{dx}\left(\frac{1}{\sigma(x)}\right) = -\frac{\sigma'(x)}{\sigma(x)^2}$, and on the other hand, $f'(x) = \frac{d}{dx}\left(1 + e^{-x}\right) = -e^{-x} = 1 - f(x) = 1 - \frac{1}{\sigma(x)} = \frac{\sigma(x)-1}{\sigma(x)}$. Equate the two expressions, and voilà!
Answer 2	Let's denote the sigmoid function as $\sigma(x) = \dfrac{1}{1+e^{-x}}$. The derivative of the sigmoid is $\dfrac{d}{dx}\sigma(x) = \sigma(x)(1 - \sigma(x))$ Here's a detailed derivation: $\begin{aligned} \frac{d}{dx}\sigma(x) &= \frac{d}{dx}\left[\frac{1}{1+e^{-x}}\right] \\ &= \frac{d}{dx}\left(1+\mathrm{e}^{-x}\right)^{-1} \\ &= -(1+e^{-x})^{-2}(-e^{-x}) \\ &= \frac{e^{-x}}{\left(1+e^{-x}\right)^2} \\ &= \frac{1}{1+e^{-x}} \cdot \frac{e^{-x}}{1+e^{-x}} \\ &= \frac{1}{1+e^{-x}} \cdot \frac{(1+e^{-x})-1}{1+e^{-x}} \\ &= \frac{1}{1+e^{-x}} \cdot \left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right) \\ &= \frac{1}{1+e^{-x}} \cdot \left(1 - \frac{1}{1+e^{-x}}\right) \\ &= \sigma(x) \cdot (1 - \sigma(x)) \end{aligned}$

Table 7: Example entry of the ARQMath dataset with preserved LaTeX formatting (post ID 78575, answer IDs 78578 and 1225116).

Problem	Simplify the following expression: $y = \dfrac{p^2 - 3p - 54}{p - 9}$
Answer/ Hints	First factor the polynomial in the numerator. $p^2 - 3p - 54 = (p-9)(p+6)$. So we can rewrite the expression as: $y = \dfrac{(p-9)(p+6)}{p-9}$. We can divide the numerator and denominator by $(p-9)$ on condition that $p \neq 9$. Therefore $y = p + 6; p \neq 9$.

Table 8: Example entry of the AMPS dataset (file `amps/khan/504/1607900679.json`).

	Original Formula	**Falsified Formula**	**Description**
Equality	$a^2 + b^2 = c^2$	$a^2 + b^2 = c^2 - 1$	Subtracted 1 from right side
		$a^2 = c^2$	Removed b^2
		$a^2 + b^{2+x} = c^2$	Inserted $+x$ in exponent of b^2
Inequality	$x > y$	$x \leq y$	Inverted $>$ to $\leq$
	$ab \leq \frac{a^2 + b^2}{2}$	$ab > \frac{a^2 + b^2}{2}$	Inverted $\leq$ to $>$
	$x \neq 0$	$x = 0$	Inverted $\neq$ to $=$
Swap	$a^2 + b^2 = c^2$	$a^2 + 2^b = c^2$	Swapped b and 2 in b^2
	$F(a) - F(b)$	$F(b) - F(a)$	Swapped order of arguments
	$\ln\left(\frac{x}{y}\right) = \ln(x) - \ln(y)$	$\ln\left(\frac{x}{y}\right) = \sin(x) - \ln(y)$	Replaced ln by sin in $\ln(x)$
	$\frac{sin(\alpha)}{a} = \frac{\sin(\beta)}{b}$	$\frac{\log(\alpha)}{a} = \frac{\sin(\beta)}{b}$	Replaced sin by log in $\sin(\alpha)$
Variable	$n! = 1 \cdot 2 \cdot \ldots \cdot n$	$k! = 1 \cdot 2 \cdot \ldots \cdot n$	Replaced n by k in $n!$
	$\sum_{i=1}^{n} i^2$	$\sum_{i=1}^{n} k^2$	Replaced i by k only in i^2
Constant	$e^{\mathrm{i}\pi} = -1$	$3^{\mathrm{i}\pi} = -1$	Replaced e by 3
		$e^{1\pi} = -1$	Replaced i by 1
		$e^{\mathrm{i}e} = -1$	Replaced π by e
		$42^{\mathrm{i}\pi} = -1$	Replaced e by 42
	$\sum_{i=1}^{\infty} \frac{1}{i^2} = \frac{\pi^2}{6}$	$\sum_{i=1}^{n} \frac{1}{i^2} = \frac{\pi^2}{6}$	Replaced ∞ by n
Distribute	$\sin(x) + \sin(y)$	$\sin(x + y)$	Applied sine additivity
	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$	$\binom{n}{k} = \frac{n!}{(k \cdot (n-k))!}$	Applied faculty multiplicity
		$\binom{n}{k} = \frac{n!}{k! \cdot (n! - k!)}$	Applied faculty multiplicity
Manual	$\forall n \in \mathbb{N} : n! = \ldots$	$\forall n \in \mathbb{R} : n! = \ldots$	Replaced $\mathbb{N}$ by $\mathbb{R}$
	$a^2 + b^2 = c^2$	$a^2 = b^2 + c^2 - 2bc\cos(\alpha)$	Similar formula
	In any right-angled triangle ...	In any right-angled square ...	Replaced “triangle” by “square”
Random	$a^2 + b^2 = c^2$	$\sin^2(\alpha) + \cos^2(\alpha) = 1$	Random formula
	In any right-angled triangle ...	The derivative of a function f is ...	Random text

Table 9: Examples of the strategies for generating falsified formulas (FalseVG).

	Symbol Groups	Typical Context	Example
Variables	a, b, c, d, e, f, g, h	Parameters	$ax^2 + bx + c = 0$
	i, j, k, l	Indices	$C_{ij} = \sum_k A_{ik} B_{kj}$
	k, l, m, n	Counts	$\binom{n}{k} = \frac{n!}{k!(n-k)!}$
	p, q, r, s, t	Parameters, Points	$x^2 + px + q = 0$
	u, v, w	Vectors	$u \times v = w$
	x, y, z	Unknowns	$x + 2y + 3z = 4$
	A, B, C, D, E, F, G, H	Matrices, Sets	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
	$Q, R, S, T, U, V, W, X, Y, Z$	Random Variables	$X = Y - Z$
	$\alpha, \beta, \gamma, \delta, \theta, \vartheta, \psi, \phi, \varphi, \rho$	Angles	$\alpha + \beta + \gamma = 180°$
	$\tau, \sigma, \lambda, \mu, \nu$	Scalars	$\lambda \left(\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right) + \mu \left(\begin{smallmatrix} y_1 \\ y_2 \end{smallmatrix}\right) = 0$
Functions	f, g, h, u, v	Generic Functions	$[uv]' = u'v + uv'$
	F, G, H, U, V	Antiderivative	$\int_a^b f(x)dx = F(b) - F(a)$
	$\tau, \sigma, \lambda, \mu, \nu$	Permutations	$\sigma \circ (\tau \circ \mu) = (\sigma \circ \tau) \circ \mu$

Table 10: Defined symbol groups for the symbol substitution of MAMUT.

string-based substitutions. As discussed earlier, a naive string replacement is inadequate for mathematical symbol substitution. For instance, if we replace `x` in `\exp{x}`, it would also unintentionally replace the occurrence of `x` within `\exp`. To address this issue, our implementation of the safe string-based substitution detects such situations resulting in a failure to avoid invalid expressions during the generation of versions ensuring high data quality. This SymPy-like expression also utilizes a predefined list of known symbols and LaTeX commands that, when present in the input, are excluded from substituting. For example, the `\exp` command is included in this list, allowing `\exp{x}` to be substituted using our safe string-based approach. This method, while being less powerful than the classical SymPy expressions, extends substitution support to a wide range of mathematical notations that can not be parsed in the classical parser. Hence, the hybrid combination of the classical SymPy expression with randomized printing and the string-based substitution, supporting a wider range of operators, aligns perfectly with our need to create a diverse, high-quality mathematical dataset with substituted symbols.

In addition, our SymPy parser implementation is adaptive. Even if an input formula can not be parsed classically, the classical parsing still succeeds for parts of it. As a result, formulas are split at delimiter symbols such as : or `\Rightarrow`. Using these extended parsing capabilities, the input `\forall x, y: x\cdot y=y\cdot x` is parsed into two sub-expressions: `\forall x, y`, which can not be parsed classically with the used implementation, and `x\cdot y=y\cdot x`, which is parsed into a classical SymPy expression. Both sub-expressions support the substitution of `x` and `y`. This results in, for instance, `\forall \alpha, a: \alpha\times a=a\times \alpha`, where randomized printing was incorporated for the right subexpression. Similarly, support for parsing entire texts containing formulas enclosed within dollar symbols, denoting the LaTeX mathematical inline mode, is integrated into the SymPy parser. To create randomized LaTeX formulas from the parsed SymPy expressions, the SymPy LaTeX printer has been enhanced to support randomized decisions. The printing process is guided by randomized settings[4], which define all the randomized decisions the printer could make. The modified SymPy code is accessible in a forked repository on GitHub[5], providing a simple interface for generating equivalent and falsified versions of a formula. Additionally, the generation code for the generated datasets based on AMPS, ARQMath, and NMFT is publicly available[6], including the logic for base formula filtering, extraction, and validation.

---

[4] anonymous

[5] anonymous

[6] anonymous

B.2 Generated Datasets

To illustrate the behavior of MAMUT and the data extraction process, we provide artificial examples for each generated dataset based on the previously shown raw data: MF in Table 11, MT in Table 12, NMF in Table 13, and MFR in Table 14. Please note that not all examples are verified as part of the actual generated datasets, but are selected to illustrate the diversity of MAMUT.

B.3 Analysis of NMF

For a better understanding of the version generation algorithms, EquVG and FalseVG, we delve into a more detailed analysis of the generated NMF dataset, visualized in Figure 5.

Figure 5a shows the distribution of strategies over the mathematical identities of the NMF dataset. The figure illustrates the proportions of how many falsified versions of a mathematical identity utilized a particular strategy. Since multiple strategies might be applied to generate a single falsified version, the proportions do not sum up to 100% per identity. Approximately half of the time, only a single strategy is applied. In general, the different strategies obviously have different proportions across the mathematical identities. The most common strategies are Variable and Swap because variables and swappable expressions (e.g., $\sin(x+y) \to \sin(x)+\sin(y)$) occur in almost all formulas, often even multiple times. About 20% of the strategies are intentionally completely random to avoid introducing a bias towards challenging negative examples. In real-world MIR applications, random pairs are more common than challenging ones. Some strategies can not be applied to certain identities, particularly the strategy Inequality, which is not applicable to most identities. The reason why some identities containing no inequalities still have a nonzero proportion for the strategy Inequality is that this strategy can be applied even after a random or manual formula (containing an inequality) is chosen as a falsified version.

Another analysis can be deducted from Figure 5b, which shows the distribution of whether a variable, function or any of them has been replaced in a generated version of a mathematical identity in the NMF dataset. Since only ten identities contain generic function symbols, substitutions of functions can only be applied to those identities regularly. Again, we recognize proportions slightly above zero for many identities not containing generic functions due to function substitutions after applying the strategies Random or Manual. The overall proportion of substituted formulas is 52.3%, but when considering only equivalent versions, this proportion rises to 81.3%. For falsified versions, the substitution proportion is about 45.1%.

Formula
$1 - \frac{1}{\sigma(x)} = \frac{\sigma(x)-1}{\sigma(x)}$
$-\dfrac{1}{\tau(y)} + 1 = \frac{1}{\tau(y)} \cdot (-1 + \tau(y))$
$\frac{1}{\nu(x)} * (\nu(x) + (-1)) = 1 - 1/\nu\,(x)$
$(\lambda(x) + (-1))/\lambda\,(x) = 1 - 1/\lambda(x)$
$1 - 1/\nu(x) = ((-1) + \nu(x))/\nu(x)$
$-1/\mu(x) + 1 = \frac{1}{\mu(x)} \cdot (\mu(x) + (-1))$
$\lambda(x) + (-1))/\lambda(x) = 1 - \frac{1}{\lambda(x)}$
$p^2 - 3p - 54 = (p-9)(p+6)$
$(p + 9 \cdot (-1)) \cdot (6 + p) = p^2 - p \cdot 3 - 54$
$p^2 - 3 \cdot p + 54 \cdot (-1) = (9 \cdot (-1) + p) \cdot (p + 6)$
$(p + 6) \cdot (-9 + p) = 54 \cdot (-1) + p^2 - p \cdot 3$
$(p-9)(p+6) = p * p - p * 3 + 54(-1)$

Table 11: Example entries for MF (based on Table 7 and Table 8).

Text
In my AI textbook there is this paragraph, without any explanation. The sigmoid function is defined as follows: "$\sigma(x) = \frac{1}{1+e^{-x}}$. This function is easy to differentiate because $\frac{d\sigma(x)}{d(x)} = \sigma(x) \cdot (1 - \sigma(x))$." It has been a long time since I've taken differential equations, so could anyone tell me how they got from the first equation to the second?
In my AI textbook there is this paragraph, without any explanation. The sigmoid function is defined as follows: "$\tau(y) = 1/(e^{-y} + 1)$. This function is easy to differentiate because $\tau(y)(-\tau(y) + 1) = \tau'(y)$." It has been a long time since I've taken differential equations, so could anyone tell me how they got from the first equation to the second?
Consider $f(x) = \dfrac{1}{\sigma(x)} = 1{+}e^{-x}$. Then, on the one hand, the chain rule gives $f'(x) = \frac{d}{dx}\left(\frac{1}{\sigma(x)}\right) = -\frac{\sigma'(x)}{\sigma(x)^2}$, and on the other hand, $f'(x) = \frac{d}{dx}\big(1 + e^{-x}\big) = -e^{-x} = 1 - f(x) = 1 - \frac{1}{\sigma(x)} = \frac{\sigma(x)-1}{\sigma(x)}$. Equate the two expressions, and voilà!
Consider $u(y) = 1/\sigma(y) = 1 + e^{-y}$. Then, on the one hand, the chain rule gives $\frac{\mathrm{d}}{\mathrm{d}y} u\,(y) = \frac{\mathrm{d}}{\mathrm{d}y} \frac{1}{\sigma(y)} = -\frac{1}{\sigma^2(y)} \frac{\mathrm{d}}{\mathrm{d}y} \sigma(y)$, and on the other hand, $u'(y) = \frac{d}{dx}\big(1 + e^{-y}\big) = -e^{-y} = 1 - u(y) = 1 - \frac{1}{\sigma(y)} = \frac{\sigma(y)-1}{\sigma(y)}$. Equate the two expressions, and voilà!
Simplify the following expression: $y = \dfrac{p^2 - 3p - 54}{p - 9}$ First factor the polynomial in the numerator. $p^2 - 3p - 54 = (p-9)(p+6)$. So we can rewrite the expression as: $y = \dfrac{(p-9)(p+6)}{p-9}$. We can divide the numerator and denominator by $(p-9)$ on condition that $p \neq 9$. Therefore $y = p + 6; p \neq 9$.
Simplify the following expression: $\frac{-54+s^2-s\cdot 3}{s-9} = z$ First factor the polynomial in the numerator. $s * s - 3 * s - 54 = (s-9) * (6+s)$. So we can rewrite the expression as: $z = \frac{1}{s-9} \times (s-9) \times (6+s)$. We can divide the numerator and denominator by $-9 + s$ on condition that $s \neq 9$. Therefore $z = s + 6; s \neq 9$.

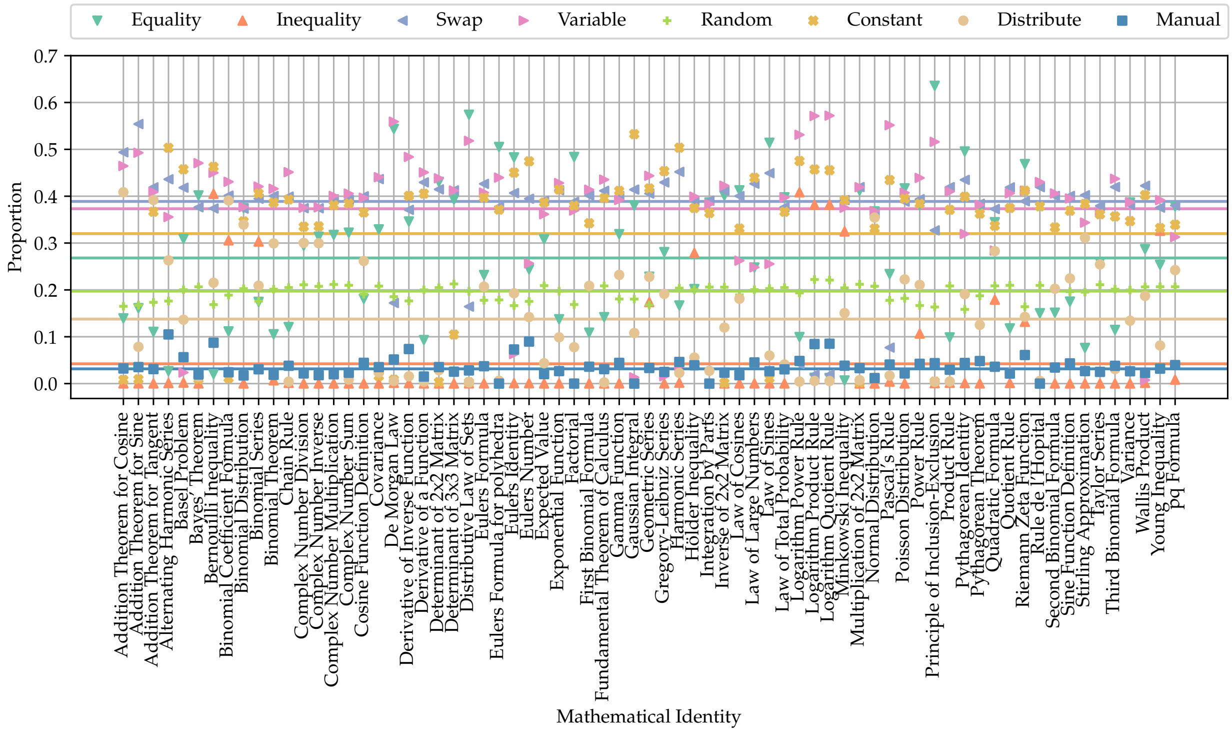
Table 12: Example entries of MT (based on Table 7 and Table 8).

Name	**Formula**	**Label**
Factorial	$d! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot \ldots \cdot d$	✓
Definition of a factorial	$\forall n \in \mathbb{N} : n! = \prod_{i=1}^{\xi} i$	✗
Definition of a factorial	$\forall n \in \mathbb{N} : (n+1)! = (n+n) \cdot n! \wedge 0! = 1$	✗
Definition of a factorial	For any natural number k we have $k!$ is defined as $k \coloneqq \prod_{j=1}^{k} j$.	✗
Factorial	For any natural number n, $n!$ can be obtained by multiplying all natural numbers from 1 to Y together.	✗
Definition of a factorial	$\forall n, j \in \mathbb{N}, n \geq j : \binom{n}{j} = \frac{1}{j! \cdot (n-j)!} \cdot n!$	✗
Factorial	$1 \cdot 2 \cdot 3 \cdot \frac{1}{4} \ldots n = n!$	✗
Factorial	$\forall m \geq 1 : m! = m \cdot (m + (-1))!, 0! = 0$	✗
Definition of a factorial	$1 * 2 * 3 * 4 \ldots x = x!$	✓
Definition of a factorial	$k! = (1-3) \cdot 18 \cdot 4 \cdot 5 / \cdot \cdots \cdot n$	✗
Factorial	$n! = \sum_{i=1}^{n} i$	✗
Factorial	The sum of two complex numbers $g_1 + \mathrm{i} \cdot h = z$ and $g_2 + \mathrm{i} \cdot f = w$ is defined as $g_1 + g_2 + i * (h + f) = w + z$.	✗
Definition of a factorial	$\theta! = 1 \cdot 2 \cdot ... \cdot \theta$	✓

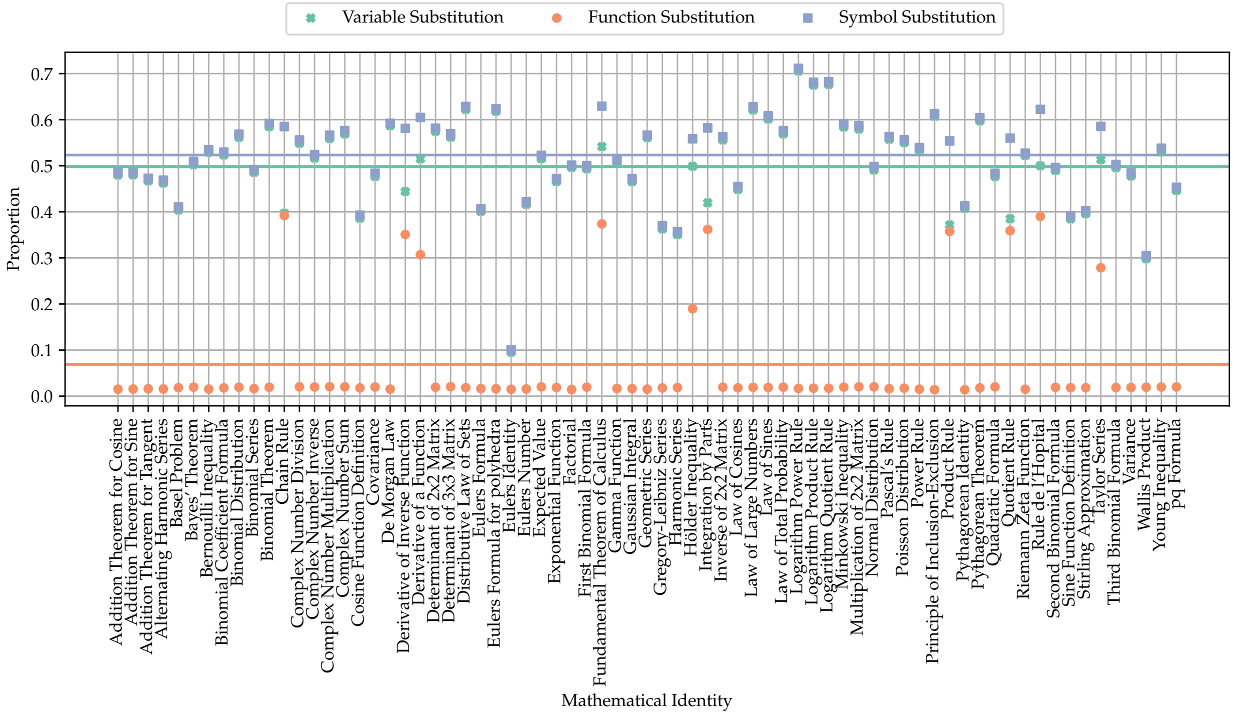
Table 13: Example entries of NMF (based on Table 4).

Formula 1	**Formula 2**	**Label**
The value of $(1+1/\tau)^{\tau}$ approaches the constant e as τ tends to infinity.	As μ approaches infinity, the expression $(1 + 1/\mu)^{\mu}$ converges to the value of $e \approx 2.718$.	✓
By utilizing the infinite series $\sum_{n=0}^{\infty} z^{1+2n} \frac{(-1)^n}{(1+2n)!}$, we can define $\sin(z)$ for all real numbers z.	For all real numbers x the expression $\sin(z)$ is defined as the infinite sum $\sum_{n=0}^{\infty} x^{2 \cdot n+1} \cdot (2 \cdot n+1)! \cdot (-1)^{-n}$.	✗
The limit as l approaches infinity of the expression $\left(1 + \frac{1}{l} \cdot y\right)^{l}$ converges to the exponential function e^{y} for any real number y.	$\forall x \in \mathbb{C} : e^x = \sum_{k=0}^{\infty} -k^x/k! = 1 + x + x^2/2! + x * x^2/3! + ...$	✗
For all real positive g with $g \neq 1$ and for all real positive s, y, we have $\log_b(sy) = \log_b(s) + \log_b(y)$.	For all real bases b such that $0 < b$ and $b \neq 1$ and for all real positive z, y, the equality $\log_b(z/y) = \log_b(z) - \log_b(y)$ holds.	✗
The derivative of a composite function $f\left(g(z)\right)$ with respect to z is given by $\frac{d}{dg(z)} f(g\left(z\right)) \cdot \frac{d}{dz} g(z)$.	The derivative of a composite function $f(g(y))$ with respect to y is given by $\frac{\mathrm{d}}{\mathrm{d}g(u)} f(g(u)) / (\frac{\mathrm{d}}{\mathrm{d}u} g(u))$.	✗
$\forall m \geq 1 : m! = m \cdot (m + (-1))!, 0! = 1$	$\forall a \in \mathbb{N} : (a+1)! = (a+1) \cdot a!, 0! = 1$	✓
Let c and b be real numbers. In this case, $(c + b)(-b + c)$ is equal to $c^2 - b^2$.	$\dfrac{1}{b-b}(a+b) = -b^2 + a^1$	✗

Table 14: Example entries of MFR.



(a) Proportions of strategies used for generating falsified versions in NMF dataset per mathematical identity.



(b) Proportion of generated (equivalent and falsified) versions with at least one variable, function, or symbol substitution (variable or function).

Figure 5: Analysis of NMF. The mean across all identities is printed as a solid line.